

Computer Architecture

Sheet (5)

- 7.9** Show a possible control sequence for implementing the instruction

MUL R1,R2

on the processor in Figure 7.1. This instruction multiplies the contents of the registers R1 and R2, and stores the result in R2. Higher-order bits in the product, if any, are discarded. Suggest additional control signals as needed, and assume that the multiplier is organized as in Figure 6.7.

- 7.10** Show the control steps for the Branch-on-Negative instruction for a processor that has the structure given in Figure 7.8.

- 7.15** Assume that the register file in Figure 7.8 is implemented as a RAM. At any given time, a location in this RAM can be accessed for either a read or a write operation. During the operation $R1 \leftarrow [R1] + [R2]$, register R1 is both a source and a destination. Explain how you would use additional latches at either the input or the output of the RAM to operate the file in a master-slave mode. Use a timing diagram to explain how your new design enables register R1 to be used as both a source and a destination in the same clock cycle.

- 7.24** Write a microroutine, such as the one shown in Figure 7.21, for the instruction

MOV X(Rsrc),Rdst

where the source and destination operands are specified in indexed and register addressing modes, respectively.

7.30 Figure P7.3 gives part of the microinstruction sequence corresponding to one of the machine instructions of a microprogrammed computer. Microinstruction B is followed

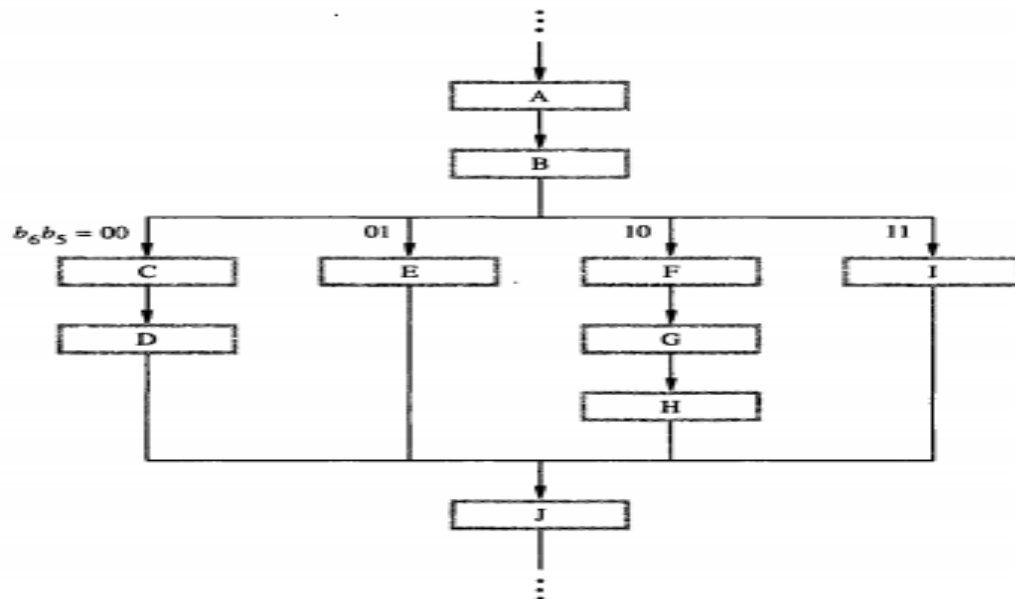


Figure P7.3 A microinstruction-sequence pattern used in Problem 7.30.

by C, E, F, or I, depending on bits b_6 and b_5 of the machine instruction register. Compare the three possible implementations described below.

- (a) Microinstruction sequencing is accomplished by means of a microprogram counter. Branching is achieved by microinstructions of the form

If b_6b_5 branch to X

where b_6b_5 is the branch condition and X is the branch address.

- (b) Same as Part a except that the branch microinstruction has the form

Branch to X, OR

where X is a base branch address. The branch address is modified by bit-ORing of bits b_5 and b_6 with the appropriate bits within X.

- (c) A field in each microinstruction specifies the address of the next microinstruction, which has bit-ORing capability.

Assign suitable addresses for all microinstructions in Figure P7.3 for each of the implementations in Parts a through c. Note that you may need to insert branch instructions in some cases. You may choose arbitrary addresses, as long as they are consistent with the method of sequencing used. For example, in Part a, you could choose addresses as follows:

Address	Microinstruction
00010	A
00011	B
00100	If $b_6b_5 = 00$ branch to XXXXX
...	...
XXXXX	C

- 7.31 It is desired to reduce the number of bits needed to encode the control signals in Figure 7.19. Suggest a new encoding that reduces the number of bits by two. How does the new encoding affect the number of control steps needed to implement an instruction?
- 7.32 Suggest a new encoding for the control signals in Figure 7.19 that reduces the number of bits needed in a microinstruction to 12. Show the effect of the new encoding on the control sequences in Figures 7.6 and 7.7.
- 7.33 Suggest a format for microinstructions, similar to Figure 7.19, if the processor is organized as shown in Figure 7.8.
- 7.34 What are the relative merits of horizontal and vertical microinstruction formats? Relate your answer to the answers to Problems 7.31 and 7.32.
- 7.35 What are the advantages and disadvantages of hardwired and microprogrammed control?